

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Daniel James Winarski, Robert George Emberty, Craig Anthony Klein,
Nils Haustein

Assignee: International Business Machines Corporation

Title: Write-Once Read-Many Hard Disk Drive Using a Worm Pointer

Serial No.: 10/824,901 Filing Date: April 14, 2004

Examiner: Daniel Tsui Group Art Unit: 2185

Docket No.: TUC920040009US1 Customer No.: 65384

Austin, Texas
April 30, 2008

Mail Stop Appeal Brief - Patents
Board of Patent Appeals and Interferences
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL APPEAL BRIEF UNDER 37 CFR § 41.37

Dear Sir:

Applicant submits this Supplemental Appeal Brief pursuant to the Notice of Appeal filed in this case on October 2, 2007 and the Notification dated April 3, 2008. The fee for the Appeal Brief was previously paid electronically via the USPTO EFS. The Board is authorized to deduct any amounts required for this supplemental appeal brief and to credit any amounts overpaid to Deposit Account No. 090449.

I. REAL PARTY IN INTEREST - 37 CFR § 41.37(c)(1)(i)

The real party in interest is the assignee, International Business Machines Corporation, as named in the caption above and as evidenced by the assignment set forth at Reel 016093, Frame 0437.

II. RELATED APPEALS AND INTERFERENCES - 37 CFR § 41.37(c)(1)(ii)

Based on information and belief, there are no appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

III. STATUS OF CLAIMS - 37 CFR § 41.37(c)(1)(iii)

Claims 1-10 and 13-32 are pending in the application. Claims 11-12 and 33-34 have been cancelled. Claims 1-10 and 13-32 stand rejected. The rejection of claims 1-10 and 13-32 is appealed. Appendix "A" contains the full set of pending claims.

IV. STATUS OF AMENDMENTS - 37 CFR § 41.37(c)(1)(iv)

No amendments after final have been requested or entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER - 37 CFR § 41.37(c)(1)(v)

The present invention, as set forth by independent claim 1, relates to a method for writing data on a data storage device (see e.g., Page 15, ¶ 0030, lines 1-3) which includes the data storage device receiving a write command (see e.g., Page 15, ¶ 0030, lines 3-5), obtaining a starting logical block addresses (LBA) and a LBA transfer length from the write command (see e.g., Page 15, ¶ 0030, lines 13-16), obtaining a first write once ready many (WORM) pointer from a WORM pointer memory, and in response to the starting LBA being greater than or equal to the WORM pointer, executing the write command (see e.g., Page 19, ¶ 0040, lines 28-32).

The present invention, as set forth by independent claim 13, relates to a data storage device which includes a data storage media for storage of data (see e.g., Page 8, ¶ 0016, lines 17-22), a processor for controlling the data storage device (see e.g., Page 10, ¶ 0020, lines 3-10), a write once ready many (WORM) pointer memory coupled to the processor for storage of a WORM pointer (see e.g., Page 10, ¶ 0020, lines 8-11), and a host device interface coupled to the processor for receiving commands from a host computer (see e.g., Page 11, ¶ 0022, lines 1-3).

The present invention, as set forth by independent claim 21 relates to a data storage system which includes a host computer (see e.g., Page 11, ¶ 0022, lines 1-3) comprising a data storage device interface, a data storage device (see e.g., Page 15, ¶ 0030, lines 1-3) which

includes a data storage media for storage of data (see e.g., Page 8, ¶ 0016, lines 17-22), a processor for controlling said data storage device (see e.g., Page 10, ¶ 0020, lines 3-10), a write once ready many (WORM) pointer memory coupled to said processor for storage of a WORM pointer (see e.g., Page 10, ¶ 0020, lines 8-11), said WORM pointer providing an inventory of locations where WORM data can be written within said data storage media (see e.g., Page 18, ¶ 0034, lines 1-11), and a host device interface coupled to said processor for sending and receiving commands with respect to said host computer (see e.g., Page 11, ¶ 0022, lines 1-3).

The present invention, as set forth by independent claim 24 relates to an article of manufacture comprising a computer readable medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform method steps for writing data on a data storage device (see e.g., Page 15, ¶ 0030, lines 1-3). The steps include said data storage device receiving a write command (see e.g., Page 15, ¶ 0030, lines 3-5), obtaining a starting logical block addresses (LBA) and a LBA transfer length from said write command (see e.g., Page 15, ¶ 0030, lines 13-16), obtaining a first write once ready many (WORM) pointer from a WORM pointer memory, said WORM pointer providing an inventory of LBAs where WORM data can be written within said data storage media (see e.g., Page 18, ¶ 0034, lines 1-11), and in response to said starting LBA being greater than or equal to said WORM pointer, executing said write command (see e.g., Page 19, ¶ 0040, lines 28-32).

VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL - 37 CFR § 41.37(c)(1)(vi)

Claims 13, 14, 19, 21 and 22 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Basham et al., U.S. Publication No. 2002/0035665 (Basham). Claims 1-5, 15, and 23-28 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Basham in view of Sokolov, U.S. Patent No. 6,018,789 (Sokolov). Claims 9, 10, 31 and 32 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Basham in view of Sokolov and further in view of Mimatsu, U.S. Publication No. 2004/0111485 (Mimatsu). Claims 16-18 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Basham in view of common practice in the art.

VII. ARGUMENT - 37 CFR § 41.37(c)(1)(vii)

Claims 13, 14, 19, 21 and 22 are Allowable Over Basham et al., U.S. Publication No. 2002/0035665 (Basham)

The present invention, as set forth by independent claim 13, relates to a data storage device which includes a data storage media for storage of data, a processor for controlling the data storage device, a write once read many (WORM) pointer memory coupled to the processor for storage of a WORM pointer, and a host device interface coupled to the processor for receiving commands from a host computer. Claim 21 relates to a data storage system of similar scope.

Basham relates to data storage subsystems that employ portable serial data storage media. In the data storage subsystem, drive level processing renders the data storage media as a write once read many (WORM) media, while permitting limited overwriting of trailing data. More specifically, the drive-level processing automatically advances a write append limiter as data is written, and prevents changes to data occurring before the write append limiter. Limited overwriting of data at the end of media is permitted since the write append limiter lags the current write location by the margin of a write allowance index.

The Examiner refers to the write append limiter 151 of Basham when setting forth that Basham discloses a WORM pointer. When discussing the write append limiter 151, Basham sets forth:

In response to the write request, the drive 108 proceeds to store the write data so as to preserve certain previously stored data, as explained below. In general, the drive 108 treats data occurring before the write append limiter as being WORM, thereby permitting overwriting of data occurring after the write append limiter. More particularly, with reference to FIG. 4, after step 406 the drive engine 110 references the write append limiter 151 and write allowance index 152 stored upon the cartridge (step 410). The write append limiter 151 identifies a sequential location on the cartridge before which data is not permitted to be altered (Basham, Para 0049, lines 3 – 12).

Accordingly, the write append limiter of Basham does not disclose or suggest a WORM pointer as claimed in claims 13 and 21

More specifically, Basham, taken alone or in combination, does not teach or suggest a data storage device which includes a data storage media for storage of data, a processor for controlling the data storage device, a write once ready many (WORM) pointer memory coupled to the processor for storage of a WORM pointer, and a host device interface coupled to the processor for receiving commands from a host computer, all as required by claim 13 and as substantially required by claim 21. Accordingly, claims 13 and 21 are allowable over Basham. Claims 14 - 20 depend from claim 13 and are allowable for at least this reason. Claims 22 and 23 depend from claim 21 and are allowable for at least this reason.

Claims 1-5, 15, and 23-28 are Allowable Over Basham et al., U.S. Publication No. 2002/0035665 (Basham) in view of Sokolov, U.S. Patent No. 6,018,789 (Sokolov).

The present invention, as set forth by independent claim 1, relates to a method for writing data on a data storage device which includes the data storage device receiving a write command, obtaining a starting logical block addresses (LBA) and a LBA transfer length from the write command, obtaining a first write once ready many (WORM) pointer from a WORM pointer memory, and in response to the starting LBA being greater than or equal to the WORM pointer, executing the write command. Claim 24 relates to an article of manufacture of similar scope.

Basham is discussed above.

Sokolov discloses a disk drive which provides a method of adaptively managing a cache segment divided into chunks by defining an unavailable data type to be stored in an element of a chunk array which indicates that the chunk is not available, and defining an available data type to be stored in an element of the chunk array that indicates the chunk is available and that indicates the number of consecutive chunks that are available. Sokolov further discloses that a logical block address uniquely references a location of data on a disk drive.

However, neither Basham or Sokolov disclose or suggest a WORM pointer which provides an inventory of LBAs where WORM data can be written within a data storage media.

More specifically, Basham and Sokolov, taken alone or in combination, do not teach or suggest a method for writing data on a data storage device which includes the data storage device receiving a write command, obtaining a starting logical block addresses (LBA) and a LBA

transfer length from the write command, obtaining a first write once ready many (WORM) pointer from a WORM pointer memory, and in response to the starting LBA being greater than or equal to the WORM pointer, executing the write command., all as required by claim 1 and as substantially required by claim 23. Accordingly, claims 1 and 24 are allowable over Basham and Sokolov. Claims 2 - 10 depend from claim 1 and are allowable for at least this reason. Claims 25 - 32 depend from claim 24 and are allowable for at least this reason.

Claims 9, 10, 31 and 32 are Allowable Over Basham in view of Sokolov and further in view of Mimatsu, U.S. Publication No. 2004/0111485 (Mimatsu)

The present invention, as set forth by claims 9 and 31, relates to sending a device type to the host computer in response to receiving a first inquiry command from a host computer,. The present invention, as set forth by claims 10 and 32, relates to sending a worm pointer to the host computer in response to receiving a second inquiry command from the host computer.

Basham and Sokolov are discussed above.

Mimatsu relates to the various functions that a disk array provides to a user. Within Mimatsu, when a function of a storage device is controlled from a computer, a common interface for controlling the function of the storage device is provided. The common interface manages an interrelationship between a storage area recognized by a host computer and a storage area provided by the storage device and associates a storage area which becomes a target of a function control instruction with the storage device that provides the storage area. Mimatsu further discloses that a disk volume is managed using a disk volume management table. Each disk volume can include device type information. (See e.g., Mimatsu ¶ 0051.)

However, Basham, Sokolov and Mimatsu do not disclose or suggest, taken alone or in combination, in response to receiving a first inquiry command from a host computer, sending a device type to the host computer, as required by claim 9 and 31 or in response to receiving a second inquiry command from the host computer, sending a worm pointer to the host computer as required by claims 10 and 32.

Claims 16-18 are Allowable Over Basham et al., U.S. Publication No. 2002/0035665 (Basham) in view of common practice in the art.

While the use of certain types of memories to store information may in fact be known in the art, the use of such a memory in combination with the method and apparatus as claimed is not disclosed or suggested by the combination of Basham with the common practice set forth by the Examiner.

VIII. CLAIMS APPENDIX - 37 CFR § 41.37(c)(1)(viii)

A copy of the pending claims involved in the appeal is attached as Appendix A.

IX. EVIDENCE APPENDIX - 37 CFR § 41.37(c)(1)(ix)

None

X. RELATED PROCEEDINGS APPENDIX - 37 CFR § 41.37(c)(1)(x)

There are no related proceedings.

XI. CONCLUSION

For the reasons set forth above, Applicant respectfully submits that the rejection of pending Claims 1-10 and 13-32 is unfounded, and requests that the rejection of claims 1-10 and 13-32 be reversed.

I hereby certify that this correspondence is being electronically submitted to the COMMISSIONER FOR PATENTS via EFS on April 30, 2008.

/Stephen A. Terrile/

Attorney for Applicant(s)

Respectfully submitted,

/Stephen A. Terrile/

Stephen A. Terrile
Attorney for Applicant(s)
Reg. No. 32,946

CLAIMS APPENDIX “A” - 37 CFR § 41.37(c)(1)(viii)

1. A method for writing data on a data storage device, comprising:
said data storage device receiving a write command;
obtaining a starting logical block addresses (LBA) and a LBA transfer length from said write command;
obtaining a first write once ready many (WORM) pointer from a WORM pointer memory, said WORM pointer providing an inventory of LBAs where WORM data can be written within said data storage media; and
in response to said starting LBA being greater than or equal to said WORM pointer, executing said write command.
2. The method of claim 1, wherein executing said write command writes said data as WORM data on said data storage device.
3. The method of claim 1, further comprising:
in response to said starting LBA being less than said first WORM pointer, aborting said write command.
4. The method of claim 1, further comprising:
in response to determining that said write command executed without errors:
calculating a second WORM pointer which is equal to the numerical sum of said first WORM pointer and said transfer length; and
storing said second WORM pointer in said WORM pointer memory.
5. The method of claim 1, further comprising:
in response to determining that said write command executed without errors:
calculating a second WORM pointer which is equal to the numerical sum of said first WORM pointer and said transfer length;

storing said second WORM pointer in said WORM pointer memory; and storing a date stamp for each said WORM pointer stored in said WORM pointer memory.

6. The method of claim 1, further comprising:
in response to determining that said write command executed with at least one error, rewriting said data.
7. The method of claim 1, further comprising:
in response to determining that said write command executed with at least one error, rewriting said data beginning at said starting LBA.
8. The method of claim 1, further comprising:
in response to determining that said write command executed with at least one error, rewriting said data beginning at an LBA that is greater than said starting LBA.
9. The method of claim 1, further comprising:
in response to receiving a first inquiry command from a host computer, sending a device type to said host computer.
10. The method of claim 9, further comprising:
in response to receiving a second inquiry command from said host computer, sending a worm pointer to said host computer.
13. A data storage device, comprising:
a data storage media for storage of data;
a processor for controlling said data storage device;
a write once ready many (WORM) pointer memory coupled to said processor for storage of a WORM pointer, said WORM pointer providing an inventory of locations where WORM data can be written within said data storage media;
a host device interface coupled to said processor for receiving commands from a host computer.

14. The data storage device of claim 13, wherein said data is stored as WORM data on said data storage media.

15. The data storage device of claim 13, wherein said processor obtains a starting logical block addresses (LBA) and a LBA transfer length from a write command received by said host device interface, obtains a first WORM pointer from said WORM pointer memory and in response to said starting LBA being greater than or equal to said first WORM pointer, executes said write command.

16. The data storage device of claim 13, wherein said WORM pointer memory is an EPROM.

17. The data storage device of claim 13, wherein said WORM pointer memory is a PROM.

18. The data storage device of claim 13, wherein said WORM pointer memory is a FLASH memory.

19. The data storage device of claim 13, wherein said WORM pointer memory is located inside a sealed portion said data storage device.

20. The data storage device of claim 13, further comprising:
a memory device for storage of a date stamp associated with each said WORM pointer.

21. A data storage system, comprising:
a host computer comprising a data storage device interface;
a data storage device comprising:
a data storage media for storage of data;
a processor for controlling said data storage device;
a write once ready many (WORM) pointer memory coupled to said processor for storage of a WORM pointer, said WORM pointer providing an inventory_

of locations where WORM data can be written within said data storage media;

a host device interface coupled to said processor for sending and receiving commands with respect to said host computer.

22. The data storage system of claim 21, wherein said data is stored as WORM data on said data storage media.

23. The data storage system of claim 21, wherein said processor obtains a starting LBA from a write command received by said host device interface, obtains a first WORM pointer from said WORM pointer memory and in response to said starting LBA being greater than or equal to said WORM pointer, executes said write command.

24. An article of manufacture comprising a computer readable medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform method steps for writing data on a data storage device, said steps comprising:

said data storage device receiving a write command;

obtaining a starting logical block addresses (LBA) and a LBA transfer length from said write command;

obtaining a first write once ready many (WORM) pointer from a WORM pointer memory, said WORM pointer providing an inventory of LBAs where WORM data can be written within said data storage media; and

in response to said starting LBA being greater than or equal to said WORM pointer, executing said write command.

25. The article of manufacture of claim 24, wherein executing said write command writes said data as WORM data on said data storage device.

26. The article of manufacture of claim 24, wherein said method steps further comprises: in response to said starting LBA being less than said first WORM pointer, aborting said write command.

27. The article of manufacture of claim 24, wherein said method steps further comprises:
in response to determining that said write command executed without errors:
calculating a second WORM pointer which is equal to the numerical sum of said first WORM pointer and said transfer length; and
storing said second WORM pointer in said WORM pointer memory.

28. The article of manufacture of claim 24, wherein said method steps further comprises:
in response to determining that said write command executed without errors:
calculating a second WORM pointer which is equal to the numerical sum of said first WORM pointer and said transfer length;
storing said second WORM pointer in said WORM pointer memory; and
storing a date stamp for each said WORM pointer stored in said WORM pointer memory.

29. The article of manufacture of claim 24, wherein said method steps further comprises:
in response to determining that said write command executed with at least one error, rewriting said data.

30. The article of manufacture of claim 24, wherein said method steps further comprises:

in response to determining that said write command executed with at least one error, rewriting said data beginning at an LBA that is greater than or equal to said starting LBA.

31. The article of manufacture of claim 24, wherein said method steps further comprises:

in response to receiving a first inquiry command from a host computer, sending a device type to said host computer.

32. The article of manufacture of claim 31, wherein said method steps further comprises:

in response to receiving a second inquiry command from said host computer, sending a worm pointer to said host computer.

EVIDENCE APPENDIX - 37 CFR § 41.37(c)(1)(ix)

None

RELATED PROCEEDINGS APPENDIX - 37 CFR § 41.37(c)(1)(x)

There are no related proceedings.